什么是静态常量和动态常量？

静态常量：编译器在编译时候会对常量进行解析，并将常量的值替换成初始化的那个值。

**动态常量（Readonly）**的值则是在运行的那一刻才获得的，编译器编译期间将其标示为只读常量，而不用常量的值代替，这样动态常量不必在声明的时候就初始化，而可以延迟到构造函数中初始化。

|  |  |
| --- | --- |
| Const | Readonly |
| 只能修饰基本数据类型，枚举，字符串 | 任何类型，包括引用 |
| 声明时直接用“ = ”初始化 | 可使用构造函数初始化 |
| Const修饰类成员变量时，不能用static修饰，但是默认是static，是类变量，仅有一份。 | Readonly修饰类成员变量时，就像普通变量一样，可以加static修饰。 |
| Const修饰的变量，值不能变。它不能修饰引用 | Readonly修饰引用，引用的指向不能变，但是指向的堆内容可以被修改！ |
| Const是静态常量，编译时标志符被实际值替换 | Readonly是动态常量，运行到变量那一刻才获得相应值 |

Const变量，编译后被嵌入到MSIL代码中，不依赖dll文件，这也是Const变量不能在构造函数中赋值的原因。Const在程序集更容易产生版本不一致问题，当dll文件的const变量被修改后，应用程序再次编译重新加载dll文件才会版本一致。

Readonly变量是在运行时加载，需请求dll文件。所以dll文件发生改变，无需重新编译应用程序，因为每次运行到Readonly变量，都会加载dll文件。